AMENDMENTS TO THE SPECIFICATION:

Please replace the paragraph beginning at page 20, line 7 with the following amended paragraph:

In the illustrated embodiment, and referring back to FIG. 2, the Operand Navigation feature is supported for the following ME register types: the GPRs 56; the next neighbor registers 74; the transfer registers 62, 64; registers or memory locations of the Local Memory 66 ~~74~~; and those of the local CSRs 70 illustrated in FIG. 3. The types of registers supported by the Operand Navigation feature can vary with the architecture of the ME or other processing element (as well as processor) to be simulated during debug. Simply put, the Operand Navigation (along with the databases that it uses) can be adapted to support any type of register of interest to a user during program debug.

Please replace the paragraph beginning at page 28, line 22 with the following amended paragraph:

The tracing algorithm ~~112~~ depends on being able to determine, for any instruction, the physical address of the registers. Once the physical addresses of the appropriate registers are determined, it is a simple matter to traverse forward or backward through Thread/PC History 134 to search for reads or writes of that register. A read or write could have occurred at a particular cycle if the ME was active and the thread state was executing (as opposed to stalled, aborted, or swapped out). When an address match is found, the cycle of interest is modified. This causes all thread windows, data and memory watch windows, and thread history windows to change to reflect the simulation state of the ME(s) at the new cycle of interest. If the matching address is found in a different context than the one where the popup menu was activated, then a Thread Window is activated for that other context.

Please replace the paragraph beginning at page 46, line 21 with the following amended paragraph:

Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, the processor 362 will receive instructions and data from a read-only memory (ROM) 364 and/or a random access memory (RAM) 366 through a CPU bus 368. A computer can generally also receive programs and data from a storage medium such as an internal disk 370 operating through a mass storage interface 372 or a removable disk 374 operating through an I/O interface 376. The flow of data over an I/O bus 378 to and from devices 370, 374 ~~274~~, (as well as input device 380 ~~280~~, and output device 282) and the processor 362 and memory 366, 364 is controlled by an I/O controller 384. User input is obtained through the input device 280, which can be a keyboard, mouse, stylus, microphone, trackball, touch-sensitive screen, or other input device. These elements will be found in a conventional desktop computer as well as other computers suitable for executing computer programs implementing the methods described here, which may be used in conjunction with output device 382 ~~282~~, which can be any display device (as shown), or other raster output device capable of producing color or gray scale pixels on paper, film, display screen, or other output medium.

Please replace the paragraph beginning at page 48, line 6 with the following amended paragraph:

Typically, the Operand Navigation tool 112 and other related processes reside on the internal disk 370 ~~374~~. These processes are executed by the processor 362 in response to a user request to the computer system's operating system in the lower-level software 105 after being loaded into memory. Any files or records produced by these processes may be retrieved from a mass storage device such as the internal disk 370 or other local memory, such as RAM 366 ~~266~~ or ROM 364.